

PEMANFAATAN METODE HEURISTIK DALAM PENCARIAN *MINIMUM SPANNING TREE* DENGAN ALGORITMA SEMUT

Alamsyah*

Abstract

Without a program, computer is just a useless box. In general, the search for the minimum spanning tree can be divided into two methods: conventional and heuristic methods. Use of heuristic methods that are expected to complete the minimum spanning tree problem with the search results are more varied and with a shorter calculation time.

In the conventional method of logic that is used only by comparing the distance of each node and then find the shortest distance. However, the weakness of conventional methods on the accuracy of the results obtained and the resulting error rate in the calculation. It would not be a problem if the data they need only a little, by contrast with the number of points that a lot will cause increased levels of calculation errors and decrease the accuracy.

The results showed that the parameters of the number of ants (m) and maximum cycle (NC_{max}) provided a major influence on the length of the cable and are consistent with the additional parameters α , β , ρ , ζ , and Q are used for the optimal solution. For the settlement in the case of graph seerhana who has 8 points with the number of ants ($m = 15$) and the maximum cycle ($NC_{max} = 10$) obtained an optimal solution to the cable length by 149 meters with a time computation 0 ms, the optimal cycle 1 and the optimal solution by 60%.

Keyword: *spanning tree, algoritma semut, heuristik*

1. Pendahuluan

Untuk menggunakan atau memfungsikan sebuah komputer maka harus terdapat program yang terdistribusi di dalamnya, tanpa program komputer hanyalah menjadi sebuah kotak yang tak berguna. Program yang terdapat pada komputer sangat bervariasi dan setiap program pasti menggunakan algoritma. Algoritma merupakan kumpulan perintah untuk menyelesaikan suatu masalah. Perintah-perintahnya dapat diterjemahkan secara bertahap dari awal hingga akhir. Masalah tersebut dapat berupa apapun dengan catatan untuk setiap masalah memiliki kriteria kondisi awal yang harus dipenuhi sebelum menjalankan algoritma. Dalam penyelesaian graf sederhana seringkali terjadi pemborosan kabel yang mengakibatkan pembengkakan biaya yang cukup besar, sehingga diperlukan ketepatan dalam pencarian *minimum spanning tree* antara titik yang satu dengan yang lainnya. Hasil pencarian *minimum spanning tree* akan menjadi pertimbangan dalam pengambilan keputusan dalam menentukan lintasan yang akan

ditempuh dan menampilkan nilai komputasi algoritma semut.

Pemanfaatan metode heuristik masih sangat jarang digunakan, sehingga dapat dirumuskan sebuah masalah yaitu dengan pemanfaatan metode heuristik diharapkan nantinya dapat menyelesaikan masalah pencarian *minimum spanning tree* dengan hasil yang lebih variatif dan dengan waktu perhitungan yang lebih singkat.

Penelitian bertujuan menyelesaikan masalah lintasan dengan menggunakan metode heuristik khususnya algoritma semut pada graf sederhana.

Manfaat yang dapat diambil dari penelitian adalah menawarkan penyelesaian yang lebih mudah dalam perhitungan (sesuai dengan tujuan algoritma heuristik) untuk pencarian *minimum spanning tree* dan dapat diaplikasikan menjadi sebuah perangkat lunak khususnya pada graf sederhana.

Dari latar belakang dan rumusan masalah yang telah dijelaskan, penelitian dibatasi pada

* Staf Pengajar Jurusan Teknik Elektro Fakultas Teknik Universitas Tadulako, Palu

algoritma yang digunakan dalam metode heuristik, yaitu algoritma semut (*Ant Algorithm*) dalam kasus graf sederhana.

2. Tinjauan Pustaka

2.1 Algoritma

Dalam dunia komputasi, istilah algoritma menjadi dasar pemikiran sebuah formulasi. Algoritma dapat didefinisikan sebagai teknik penyusunan langkah-langkah penyelesaian masalah dalam bentuk kalimat dengan jumlah kata terbatas tetapi tersusun secara logis dan sistematis. Kalimat-kalimat ini dapat diterjemahkan secara bertahap dari awal hingga akhir. Algoritma harus berhenti setelah mengerjakan serangkaian tugas atau langkahnya terbatas dan setiap langkah harus didefinisikan dengan tepat sehingga tidak memiliki arti ganda [Suarga, 2006].

Algoritma sering mempunyai langkah pengulangan (*iterasi*) atau memerlukan keputusan (*logika Boolean* dan *perbandingan*) sampai tugasnya selesai. Desain dan analisis algoritma adalah suatu cabang khusus dalam ilmu komputer yang mempelajari karakteristik dan performansi dari suatu algoritma dalam menyelesaikan masalah. Algoritma pun bersifat bebas, artinya tidak bergantung pada sistem komputer atau bahasa pemrograman yang digunakan. Algoritma yang berbeda dapat diterapkan pada suatu masalah dengan kriteria yang sama. Dalam penyelesaian masalah, ukuran banyak komputasi dari suatu **algoritma** dinyatakan dalam kompleksitas. Jika sebuah permasalahan dalam diselesaikan dalam waktu yang singkat dikatakan kompleksitas algoritma rendah, sementara algoritma yang membutuhkan waktu lama untuk menyelesaikan masalahnya mempunyai kompleksitas yang tinggi [Suarga, 2006].

2.2 Graf

Graf adalah kumpulan simpul (*nodes*) yang dihubungkan satu sama lain melalui sisi/busur (*edges*) [Buckley, 1990]. Suatu Graf G terdiri dari dua himpunan yaitu himpunan V dan himpunan E .

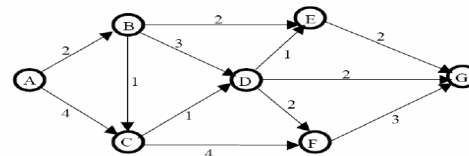
- Verteks (simpul) : V = himpunan simpul yang terbatas dan tidak kosong
- Edge (sisi/busur): E = himpunan busur yang menghubungkan sepasang simpul.

Simpul-simpul pada graf dapat merupakan obyek sembarang seperti kota, atom-atom suatu zat,

nama anak, jenis buah, komponen alat elektronik dan sebagainya. Busur dapat menunjukkan hubungan (relasi) sembarang seperti rute penerbangan, jalan raya, sambungan telepon, ikatan kimia, dan lain-lain. Notasi graf: $G(V,E)$ artinya graf G memiliki V simpul dan E busur.

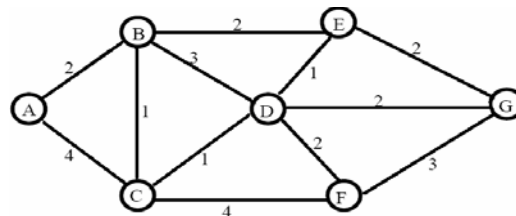
Menurut arah dan bobotnya, graf dibagi menjadi empat bagian, yaitu:

- Graf berarah dan berbobot : tiap busur mempunyai anak panah dan bobot. Gambar 1. menunjukkan graf berarah dan berbobot yang terdiri dari tujuh titik yaitu titik A,B,C,D,E,F,G. Titik menunjukkan arah ke titik B dan titik C, titik B menunjukkan arah ke titik D dan titik C, dan seterusnya. Bobot antar titik A dan titik B pun telah di ketahui.



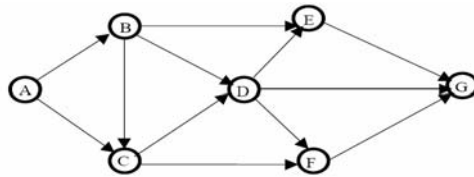
Gambar 1. Graf berarah dan berbobot

- Graf tidak berarah dan berbobot : tiap busur tidak mempunyai anak panah tetapi mempunyai bobot. Gambar 2 menunjukkan graf tidak berarah dan berbobot. Graf terdiri dari tujuh titik yaitu titik A,B,C,D,E,F,G. Titik A tidak menunjukkan arah ke titik B atau C, namun bobot antara titik A dan titik B telah diketahui. Begitu juga dengan titik yang lain.



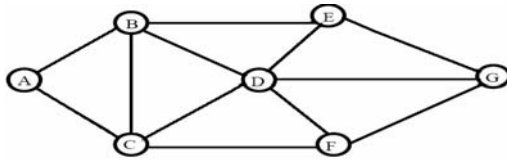
Gambar 2. Graf tidak berarah dan berbobot

- Graf berarah dan tidak berbobot: tiap busur mempunyai anak panah yang tidak berbobot. Gambar 3. menunjukkan graf berarah dan tidak berbobot.



Gambar 3. Graf berarah dan tidak berbobot

- d. Graf tidak berarah dan tidak berbobot: tiap busur tidak mempunyai anak panah dan tidak berbobot.



Gambar 4.. Graf tidak berarah dan tidak berbobot

2.3 Pohon Rentang Minimum (*Minimum Spanning Tree*)

Suatu masalah pohon rentang minimum (*minimum spanning tree*) menyangkut suatu himpunan nodes dan suatu himpunan cabang yang diusulkan (*proposed*), yang satupun tidak ada yang terorientasi. Tiap-tiap cabang yang diusulkan memiliki suatu biaya tak-negatif yang berkaitan dengannya. Tujuannya adalah menyusun suatu jaringan tersambung yang mengandung semua nodes dan sedemikian rupa sehingga jumlah biaya yang berkaitan dengan cabang-cabang yang benar digunakan adalah minimum.

Minimum Spanning Tree merupakan variasi dari persoalan rute terpendek yang perbedaannya terletak pada lintasan yang dicari [Dimiyati, 1987]. Pada rute terpendek dicari lintasan/rute dari sumber ke tujuan yang memberi total jarak minimum, sedangkan pada *Minimum Spanning Tree* ini yang dipersoalkan adalah menentukan busur-busur yang menghubungkan nodes yang ada pada pada jaringan, sehingga diperoleh panjang busur total minimum.

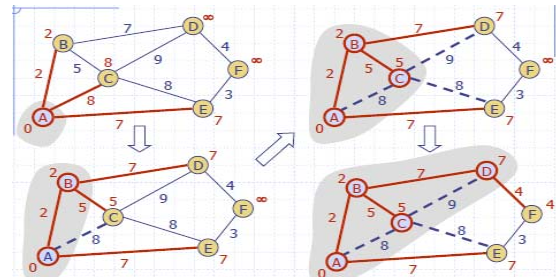
Sedangkan persamaan pada kedua kasus ini adalah suatu jaringan tak terarah, dengan informasi yang mencakup nodes-nodes yang

beserta jarak berupa biaya, waktu dan besaran lainnya antar nodes.

Contoh-contoh praktis dari persoalan spanning tree antara lain :

- Perencanaan jaringan transportasi. Dalam hal ini *nodes*nya merupakan terminal, sedangkan busur-busurnya dapat berupa jalan raya. Persoalan ialah menentukan pola transportasi yang dapat melayani seluruh terminal dengan jarak minimum
- Perencanaan jaringan komunikasi berskala besar
- Perencanaan jaringan distribusi

Persoalan *Spanning Tree* ini dapat diselesaikan dengan cara mudah sebagai berikut: memilih sembarang salah satu *nodes*, kemudian menghubungkan *nodes* tersebut dengan *nodes* lain yang terdekat, menentukan *nodes* lain yang belum terhubung, jarak yang paling dekat dengan *nodes* yang sudah terhubung pada langkah sebelumnya, kemudian menghubungkan *nodes* ini. mengulangi langkah ini sehingga seluruh *nodes* dapat terhubung.



Gambar 5. Contoh *minimum spanning tree*

Secara umum penyelesaian masalah pencarian *minimum spanning tree* dapat dilakukan menggunakan dengan dua buah metode, yaitu metode algoritma konvensional dan metode heuristik. Metode algoritma konvensional diterapkan dengan cara perhitungan matematis seperti biasa, sedangkan metode heuristik diterapkan dengan perhitungan kecerdasan buatan, dengan menentukan basis pengetahuan dan perhitungannya.

- Metode konvensional

Metode konvensional berupa algoritma yang menggunakan perhitungan matematis biasa.

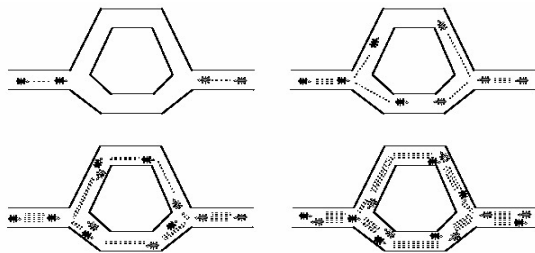
Ada beberapa metode konvensional yang biasa digunakan untuk melakukan pencarian *minimum spanning tree*, diantaranya algoritma Prim dan Kruskal.

b. Metode heuristik

Adalah sub bidang dari kecerdasan buatan yang digunakan untuk melakukan pencarian dan penentuan rute terpendek. Ada beberapa algoritma pada metode heuristik yang biasa digunakan dalam pencarian *minimum spanning tree*. Namun dalam penelitian dibatasi hanya membahas algoritma semut.

2.4 Algoritma semut

Algoritma Semut diadopsi dari perilaku koloni semut yang dikenal sebagai sistem semut (Dorigo, 1996). Secara alamiah koloni semut mampu menemukan rute terpendek dalam perjalanan dari sarang ke tempat-tempat sumber makanan.



Gambar 6. Perjalanan semut ke sumber makanan

Koloni semut dapat menemukan rute terpendek antara sarang dan sumber makanan berdasarkan jejak kaki pada lintasan yang telah dilalui. Semakin banyak semut yang melalui suatu lintasan, maka semakin jelas bekas jejak kakinya. Hal ini menyebabkan lintasan yang dilalui semut dalam jumlah sedikit, semakin lama semakin berkurang kepadatan semut yang melewatinya, atau bahkan akan tidak dilewati sama sekali. Sebaliknya lintasan yang dilalui semut dalam jumlah banyak, semakin lama akan semakin bertambah kepadatan semut yang melewatinya, atau bahkan semua semut melalui lintasan tersebut.

Gambar 6 menunjukkan perjalanan semut dalam menemukan jalur terpendek dari sarang ke sumber makanan, terdapat dua kelompok semut yang melakukan perjalanan. Kelompok semut L berangkat dari arah kiri ke kanan dan kelompok

semut R berangkat dari kanan ke kiri. Kedua kelompok berangkat dari titik yang sama dan dalam posisi pengambilan keputusan jalan sebelah mana yang akan diambil. Kelompok L membagi dua kelompok lagi. Sebagian melalui jalan atas dan sebagian melalui jalan bawah. Hal ini juga berlaku pada kelompok R. Gambar 6.b dan Gambar 6.c menunjukkan bahwa kelompok semut berjalan pada kecepatan yang sama dengan meninggalkan *feromon* atau jejak kaki di jalan yang telah dilalui. *Feromon* yang ditinggalkan oleh kumpulan semut yang melalui jalan atas telah mengalami banyak penguapan karena semut yang melalui jalan atas berjumlah lebih sedikit dari pada jalan yang di bawah. Hal ini disebabkan jarak yang ditempuh lebih panjang daripada jalan bawah. Sedangkan *feromon* yang berada di jalan bawah penguapannya cenderung lebih lama. Karena semut yang melalui jalan bawah lebih banyak daripada semut yang melalui jalan atas. Gambar 6.d menunjukkan bahwa semut-semut yang lain pada akhirnya memutuskan untuk melewati jalan bawah karena *feromon* yang ditinggalkan masih banyak. Sedangkan *feromon* pada jalan atas sudah banyak menguap sehingga semut-semut tidak memilih jalan atas tersebut. Semakin banyak semut yang melalui jalan maka semakin sedikit semut yang melalui jalan, maka *feromon* yang ditinggalkan semakin berkurang bahkan hilang. Dari sinilah kemudian terpilihlah jalur terpendek antara sarang dan sumber makanan.

Dalam algoritma semut, diperlukan beberapa variabel dan langkah-langkah untuk menentukan jalur terpendek, yaitu:

Langkah 1:

a. Inisialisasi harga parameter-parameter algoritma. Parameter-parameter yang di inisialisasikan adalah:

- 1). Intensitas jejak semut antar kota dan perubahannya (τ_{ij})
- 2). Banyak kota (n) termasuk x dan y (koordinat) atau d_{ij} (jarak antar kota)
- 3). Tetapan siklus-semut (Q)
- 4). Tetapan pengendali intensitas jejak semut (α)
- 5). Tetapan pengendali visibilitas (β)
- 6). Visibilitas antar kota = $1/d_{ij}$ (η_{ij})
- 7). Banyak semut (m)
- 8). Tetapan penguapan jejak semut (ρ)
- 9). Jumlah siklus maksimum (NC_{max}) bersifat tetap selama algoritma dijalankan, sedangkan τ_{ij} akan selalu diperbaharui

harganya pada setiap siklus algoritma mulai dari siklus pertama (NC=1) sampai tercapai jumlah siklus maksimum (NC=NCmax) atau sampai terjadi konvergensi.

b. Inisialisasi kota pertama setiap semut.

Setelah inisialisasi τ_{ij} dilakukan, kemudian m semut ditempatkan pada kota pertama tertentu secara acak.

Langkah 2:

Pengisian kota pertama ke dalam tabu list. Hasil inisialisasi kota pertama setiap semut dalam langkah 1 harus diisikan sebagai elemen pertama tabu list. Hasil dari langkah ini adalah terisinya elemen pertama tabu list setiap semut dengan indeks kota tertentu, yang berarti bahwa setiap tabuk(1) bisa berisi indeks kota antara 1 sampai n sebagaimana hasil inisialisasi pada langkah 1.

Langkah 3:

Penyusunan rute kunjungan setiap semut ke setiap kota. Koloni semut yang sudah terdistribusi ke sejumlah atau setiap kota, akan mulai melakukan perjalanan dari kota pertama masing-masing sebagai kota asal dan salah satu kota-kota lainnya sebagai kota tujuan. Kemudian dari kota kedua masing-masing, koloni semut akan melanjutkan perjalanan dengan memilih salah satu dari kota-kota yang tidak terdapat pada tabuk sebagai kota tujuan selanjutnya. Perjalanan koloni semut berlangsung terus menerus sampai semua kota satu persatu dikunjungi atau telah menempati tabuk. Jika s menyatakan indeks urutan kunjungan, kota asal dinyatakan sebagai tabuk(s) dan kota-kota lainnya dinyatakan sebagai {N-tabuk}, maka untuk menentukan kota tujuan digunakan persamaan probabilitas kota untuk dikunjungi sebagai berikut:

$$P_{ij}^k = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k' \in \{N - \text{tabu}_k\}} [\tau_{ik'}]^\alpha \cdot [\eta_{ik'}]^\beta} \dots\dots\dots(1)$$

$$j \in \{N - \text{tabu}_k\}$$

$$P_{ij}^k = 0, \text{ untuk } j \text{ lainnya.}$$

dengan i sebagai indeks kota asal dan j sebagai indeks kota tujuan.

Langkah 4:

- a. Perhitungan panjang rute setiap semut. Perhitungan panjang rute tertutup (length closed tour) atau L_k setiap semut dilakukan setelah satu siklus diselesaikan oleh semua semut. Perhitungan dilakukan berdasarkan tabuk masing-masing dengan persamaan berikut:

$$L_k = \sum_{s=1}^{n-1} d_{\text{tabuk}(s), \text{tabuk}(s+1)} \dots\dots\dots(2)$$

dengan d_{ij} adalah jarak antara kota i ke kota j yang dihitung berdasarkan persamaan:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

b. Pencarian rute terpendek.

Setelah L_k setiap semut dihitung, akan didapat harga minimal panjang rute tertutup setiap siklus atau $L_{\min NC}$ dan harga minimal panjang rute tertutup secara keseluruhan adalah atau L_{\min} .

c. Perhitungan perubahan harga intensitas jejak kaki semut antar kota.

Koloni semut akan meninggalkan jejak-jejak kaki pada lintasan antar kota yang dilaluinya. Adanya penguapan dan perbedaan jumlah semut yang lewat, menyebabkan kemungkinan terjadinya perubahan harga intensitas jejak kaki semut antar kota. Persamaan perubahan ini adalah:

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k \dots\dots\dots(3)$$

dengan $\Delta \tau_{ij}$ adalah perubahan harga intensitas jejak kaki semut antar kota setiap semut yang dihitung berdasarkan persamaan :

$$\Delta \tau_{ij}^k = \frac{Q}{L_k} \text{ untuk } (i,j) \in \text{kota asal dan kota tujuan dalam } \text{tabu}_k$$

$$\Delta \tau_{ij}^k = 0 \text{ untuk } (i,j) \text{ lainnya.}$$

Langkah 5:

- Perhitungan harga intensitas jejak kaki semut antar kota untuk siklus selanjutnya. Harga intensitas jejak kaki semut antar kota pada semua lintasan antar kota ada kemungkinan berubah karena adanya penguapan dan perbedaan jumlah semut yang melewati. Untuk siklus selanjutnya, semut yang akan melewati lintasan tersebut harga intensitasnya telah berubah. Harga intensitas jejak kaki semut antarkota untuk siklus selanjutnya dihitung dengan persamaan:

$$\tau_{ij} = \rho \cdot \tau_{ij} + \Delta \tau_{ij} \dots\dots\dots(4)$$

- Atur ulang harga perubahan intensitas jejak kaki semut antar kota. intensitas jejak semut antar kota perlu diatur kembali agar memiliki nilai sama dengan nol.

Langkah 6:

Pengosongan tabu list, dan ulangi langkah 2 jika diperlukan. Tabu list perlu dikosongkan untuk diisi lagi dengan urutan kota yang baru pada siklus selanjutnya, jika jumlah siklus maksimum belum tercapai atau belum terjadi konvergensi. Algoritma diulang lagi dari langkah 2 dengan harga parameter intensitas jejak kaki semut antar kota yang sudah diperbaharui.

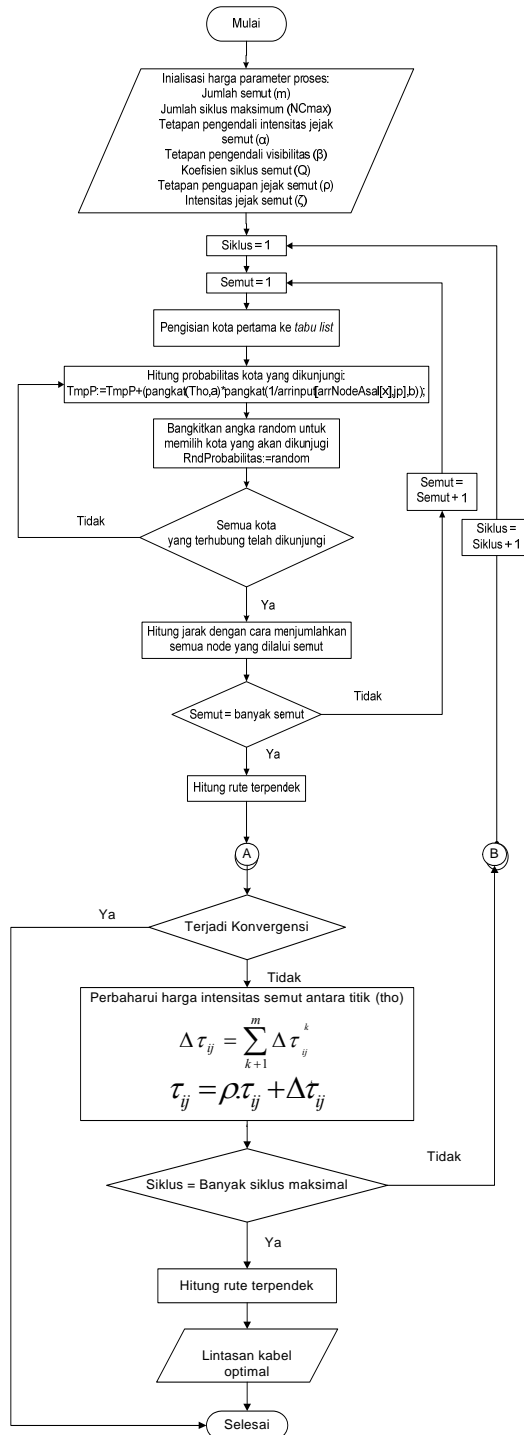
2.5 Flowchart Algoritma Semut

Flowchart Algoritma semut disajikan pada Gambar 7.

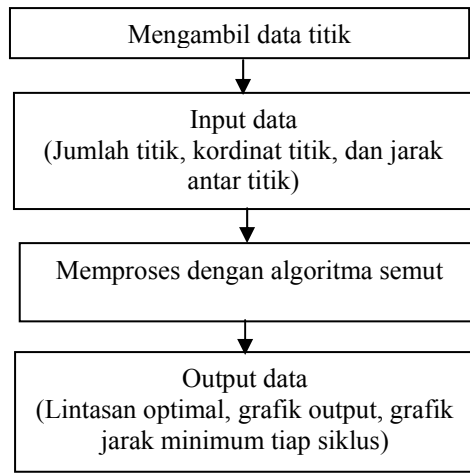
2.6 Kerangka konseptual

Penelitian dimulai dari pengambilan titik yang merupakan titik asal sampai ke titik tujuan beserta lintasan yang menghubungkannya. Karena semua titik akan digunakan, maka dilakukan proses seleksi lintasan yang nantinya akan diproses secara pola semut. Selanjutnya lintasan ini akan diproses secara *semut* hingga diperoleh lintasan optimal dan waktu komputasi tertentu. Gambaran umum

dari kerangka konseptual penelitian ini dapat diilustrasikan pada gambar 8.



Gambar 7. Flowchart Algoritma Semut



Gambar 8. Kerangka Konseptual Penelitian

3. Metode Penelitian

3.1 Rancangan penelitian

Pada penelitian ini, terdapat beberapa metode pengumpulan data yang digunakan, yaitu:

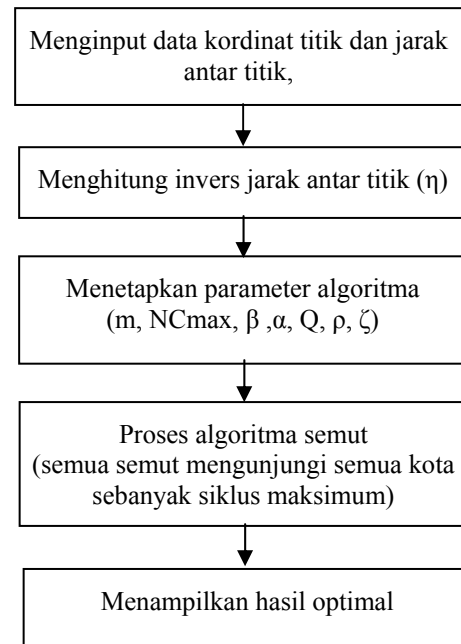
- Metode kepustakaan
Metode pengumpulan data kepustakaan dilakukan dengan mengumpulkan data-data dari sumber atau buku yang relevan terhadap penelitian.
- Metode wawancara
Metode wawancara dilakukan dengan cara tatap muka dan menanyakan langsung kepada objek yang pernah melakukan penelitian sebelumnya.
- Penyusunan Algoritma
Menentukan spesifikasi sistem yang akan digunakan dalam hal ini penyusunan algoritma semut dengan metode pencarian minimum spanning tree.
- Pembuatan Program Aplikasi
Pada tahap ini dibuat program yang menggunakan algoritma semut dengan didukung oleh perangkat lunak Delphi 7.0

3.2 Jenis penelitian

Penelitian ini adalah penelitian kualitatif dengan melakukan eksperimen dalam menganalisis dan merancang perangkat lunak untuk menyelesaikan masalah pencarian *minimum spanning tree* dengan menggunakan algoritma semut pada kasus graf sederhana.

3.3 Analisa sistem

Algoritma Semut melalui tahap-tahap kerja, yaitu menginput data kordinat titik dan jarak antar titik, selanjutnya menghitung invers jarak antar titik kemudian menetapkan parameter algoritma (jumlah semut, jumlah siklus maksimum, pengendali visibilitas, pengendali intensitas jejak semut, penguapan jejak semut, intensitas jejak semut), proses algoritma semut (semua semut mengunjungi semua kota sebanyak siklus maksimum), menampilkan hasil optimal. Seperti yang tampak pada gambar 9.



Gambar 9. Bagan kerja algoritma semut

3.4 Metode Pengukuran

Penulis melakukan percobaan dengan menggunakan tujuh buah parameter yang ditentukan sendiri. Parameter-parameter tersebut adalah jumlah semut, jumlah siklus maksimum, pengendali visibilitas, pengendali intensitas jejak semut, penguapan jejak semut, intensitas jejak semut. Ketujuh parameter tersebut diinisiasikan dengan batasan sebagai berikut:

- Banyaknya semut 1 - 100
- Siklus maksimal 1 - 10
- Tetapan pengendali visibilitas 0 - 1
- Tetapan pengendali intensitas jejak semut 0 - 1

- e. Tetapan siklus semut 0 - 1
- f. Tetapan penguapan jejak semut 0 - 1
- g. Intensitas jejak semut antar titik 0 - 1

Banyaknya semut, siklus maksimum, tetapan pengendali visibilitas, tetapan pengendali intensitas jejak semut dan tetapan penguapan jejak semut bernilai tetap untuk semua siklus, sedangkan intensitas jejak semut nilainya berbeda untuk setiap siklus.

Setelah diperoleh solusi optimal (semut telah melakukan proses sebanyak siklus maksimum maka proses dihentikan dan hasil optimal ditampilkan).

4. Hasil dan Pembahasan

Pada dasarnya permasalahan pencarian *minimum spanning tree* antar titik merupakan pencarian rute antar titik yang telah diketahui titik koordinat atau jarak. Dengan mengetahui konsep pencarian pencarian *minimum spanning tree* antar titik, untuk selanjutnya dapat diterapkan pada kasus graf sederhana pada berbagai rute yang ingin diketahui panjang kabel yang dibutuhkan. Contoh kasus yang akan diambil adalah seperti terlihat pada gambar 8 dengan kasus graf sederhana 8 titik. Terdapat dua jenis kasus yang bisa diturunkan dari gambar di atas. Kasus pertama adalah mengetahui

jarak antar node yang ditunjukkan dengan garis penghubung antar titik. Kasus yang kedua adalah dengan mengetahui koordinat titik saja. Gambar 10 merupakan jenis kasus yang pertama yaitu dengan mengetahui jarak antar titik. Untuk kasus pertama, penyelesaian cenderung lebih mudah karena jarak antar titik telah diketahui sebagai berikut:

Tabel 1. Jarak antar titik

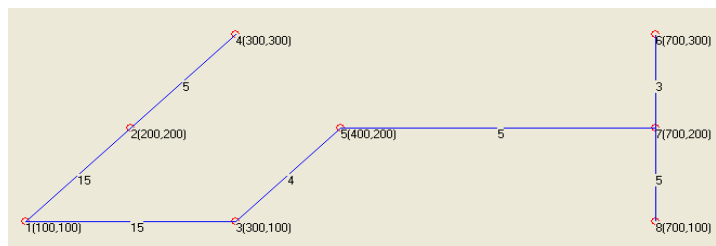
Titik	1	2	3	4	5	6	7	8
1	0	15	15	0	0	0	0	0
2	15	0	18	5	0	0	0	0
3	15	18	0	0	4	0	0	15
4	0	5	0	0	15	15	0	0
5	0	0	4	15	0	0	5	0
6	0	0	0	15	0	0	3	0
7	0	0	0	0	5	3	0	5
8	0	0	15	0	0	0	5	0

diberikan 15 semut buatan dan menggunakan harga parameter $NC_{max} = 10$, $\alpha = 1$, $\beta = 1$, $\rho = 1$, dan $\zeta = 1$.

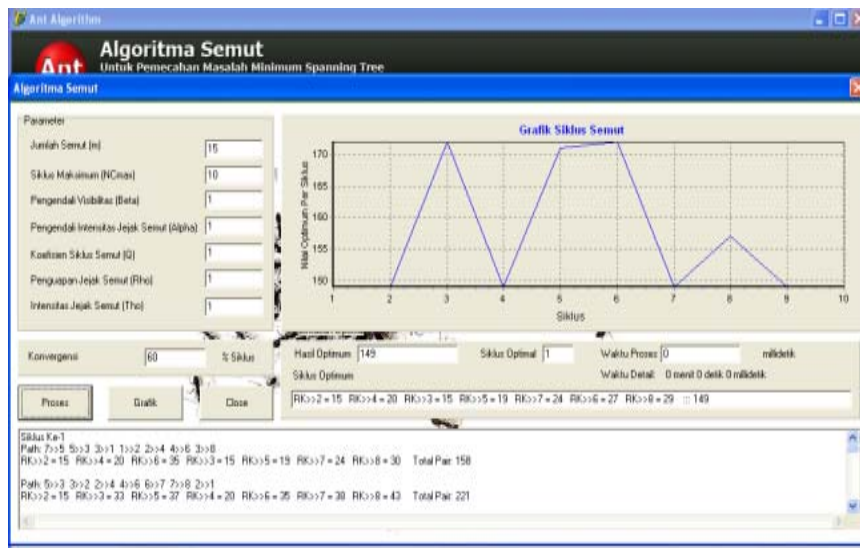
Dari hasil 20 kali proses algoritma semut diperoleh panjang kabel optimal 149 meter, waktu komputasi 15 ms, solusi optimal sebesar 60%, siklus optimal 1 serta lintasan optimal “4>>2 2>>1 1>>3 3>>5 5>>7 7>>6 7>>8” seperti yang terlihat pada gambar 11 dan gambar 12.

Tabel 2. Invers jarak antar titik

Titik	1	2	3	4	5	6	7	8
1	0	0.07	0.07	0	0	0	0	0
2	0.07	0	0.06	0.20	0	0	0	0
3	0.07	0.06	0	0	0.25	0	0	0.07
4	0	0.20	0	0	0.07	0.07	0	0
5	0	0	0.25	0.07	0	0	0.20	0
6	0	0	0	0.07	0	0	0.33	0
7	0	0	0	0	0.20	0.33	0	0.20
8	0	0	0.07	0	0	0	0.20	0



Gambar 11. Lintasan optimal hasil proses algoritma semut



Gambar 12. Siklus optimal dan waktu komputasi

5. Kesimpulan dan Saran

5.1 Kesimpulan

- Pemanfaatan teknologi informasi pada pencarian *minimum spanning tree* menghasilkan suatu hasil atau keluaran yang akurat dan tepat, untuk kasus perencanaan jaringan telepon pada graf sederhana dengan mempertimbangkan beberapa parameter yang lain.
- Untuk kasus yang berbeda algoritma akan memberikan hasil yang berbeda, tidak dapat dipastikan bahwa algoritma semut lebih baik dibanding dengan algoritma lain (heuristik).
- Dengan metode heuristik, penyelesaian kasus pada perencanaan jaringan kabel telepon pada graf sederhana yang memiliki 8 titik dengan memberikan jumlah semut ($m=15$) dan siklus maksimum ($NCmax = 10$) diperoleh panjang kabel optimal sebesar 149 meter, waktu komputasi 0 ms, siklus optimal 1 dan solusi optimal sebesar 60 %.
- Secara konsep algoritma, metode konvensional lebih mudah untuk dipahami, namun hasil yang diperoleh dari metode heuristik lebih variatif.

5.2 Saran

- Diharapkan ada penelitian lebih lanjut untuk mengetahui efisiensi dari pencarian *minimum spanning tree* menggunakan metode heuristik.

- Diharapkan adanya penelitian yang dapat membandingkan antar metode heuristik yang lain.

6. Daftar Pustaka

- Bucklely, Fred and Frank Harary. 1990. *Distance in Graph*, Addison Wesley Publishing Company.
- Capriles, Priscila V. S. Z., Fonseca, Leonardo Goliatt da dan Barbosa, Helio J.C. *Ant Colony Algorithms Applied To Discrete Optimization Problems*. http://200.231.172.253/cnmac/storal2/leonardo_fonseca_ST1_8.pdf (akses terakhir pada tanggal 2 Oktober 2009.)
- Deo, Marsingh. 2001. *Graph Theory Application to Engineering and Computer Science*. Prentice Hall Inc. New Delhi
- Dorigo. 1996. The Ant System: Optimization by a colony of cooperating agents, IEEE Transaction System, Man and Cybernetics-Part B, Vol.26, No. 1, pp.1-13
- Kusumadewi, S., H., Purnomo. 2005. *Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik*. Graha Ilmu, Yogyakarta

- Munir, Rinaldi. 2005. *Matematika Diskrit*. Informatika, Bandung
- Munir, Rinaldi. 2007. *Algoritma dan Pemrograman*. Informatika, Bandung
- Rosen, Kenneth H. 1991. *Discrete mathematics and its application 2nd*. Vaga New York.
- Suarga. 2006. *Algoritma Pemrograman*. Andi, Yogyakarta

Dari hasil pengujian secara keseluruhan terlihat